

Universidad De San Carlos de Guatemala
Facultad de Ingeniería
Escuela de Ciencias y Sistemas

Lenguajes Formales y de Programación
Sección "B-"



MANUAL TÉCNICO:

Pixel art

Estudiante: José Ernesto Pajoc Raymundo

Carné: 201115455

Segundo semestre del 2,021

Presentación del proyecto: 24/09/2021

Objetivos

General:

Presentar las clases, métodos, funciones, librerías y diagramas que conforman al proyecto para el funcionamiento adecuado del software.

Específicos:

- Mostrar la descripción sobre la plataforma en la cual fue desarrollado el software.
- Identificar las librerías, funciones, métodos y atributos que conforman la parte principal del programa.
- Presentar el flujo del software a través de daigramas.

Introducción

El programa corresponde al proyecto 1 del curso Lenguajes Formales y de Programación, el cual consiste en desarrollar una aplicación en el lenguaje de programación Python, con el cual se carga un archivo de texto plano a la memoria para luego ser analizado por un autómata finito implementado por el programador, obteniendo de esa forma los parámetros necesarios para poder generar las imágenes en píxeles y poderlas mostrar al usuario en un entorno gráfico como también en archivos HTML donde se describe el tipo de filtro que se le aplica a cada imagen.

Información técnica

El software se desarrolló utilizando las siguientes tecnologías:

- Sistema operativo: Windows 10 (64 bits).
- Lenguaje utilizado: Python 3.8
- IDE: Visual Code, Versión: 1.56.2
- Navegador web: Google Chrome Versión 92.0.4515.159

Descripción del problema

La empresa Bitxelart se dedica a la elaboración de imágenes digitales en estilo pixel art, para realizar estas piezas digitales un empleado toma una imagen que utilizará como referencia y la divide en una cuadrícula, anotando en un archivo de texto, con extensión pxa, las características de la misma (color, dimensiones, etc), luego este archivo es trasladado a un diseñador digital que, utilizando el lenguaje de marcado HTML, dibuja la imagen solicitada.

Se debe realizar un programa que sea capaz de leer los archivos de texto de estas piezas y genere las imágenes correspondientes, para así automatizar este proceso.

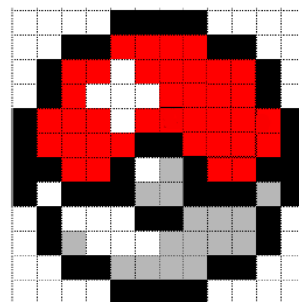
Ejemplo: archivo de texto.

```
TITULO="Pokebola";
ANCHO=300;
ALTO=300;
FILAS=12;
COLUMNAS=12;
CELDA = {
    [0,0,FALSE,#000000],
    [0,1,FALSE,#000000],
    [3,3,FALSE,#000000],
    [3,4,TRUE,#000000],
    [3,5,TRUE,#000000],
    [3,6,TRUE,#000000],
    [3,7,TRUE,#000000],
    [4,1,FALSE,#000000]
};
FILTROS = MIRRORX;

####

TITULO="Estrella";
ANCHO=300;
ALTO=300;
FILAS=4;
COLUMNAS=4;
CELDA = {
    [0,0,FALSE,#000000],
    [1,1,FALSE,#000000],
    [3,3,FALSE,#000000],
    [2,1,FALSE,#000000]
};
FILTROS = MIRRORX,MIRROR,Y,DOUBLEMIRROR;
```

Ejemplo: imagen a producir.



Descripción de la Solución

Para poder desarrollar el software se analizo primero la información y el tipo de archivo que se debe cargar al sistema, con ello se crearon expresiones regulares las cuales permiten extraer los datos de la imagen, para continuar con el proceso de la creación del árbol binario y luego dar paso a la elaboración de un autómata finito determinista el cual es programado dentro de la aplicación.

Donde:

L = [a-zA-ZñÑ]

D = [0-9]

I = [carácter ASCII imprimible en el rango de 128 hasta 239]

Tabla I

Tokens implementados.

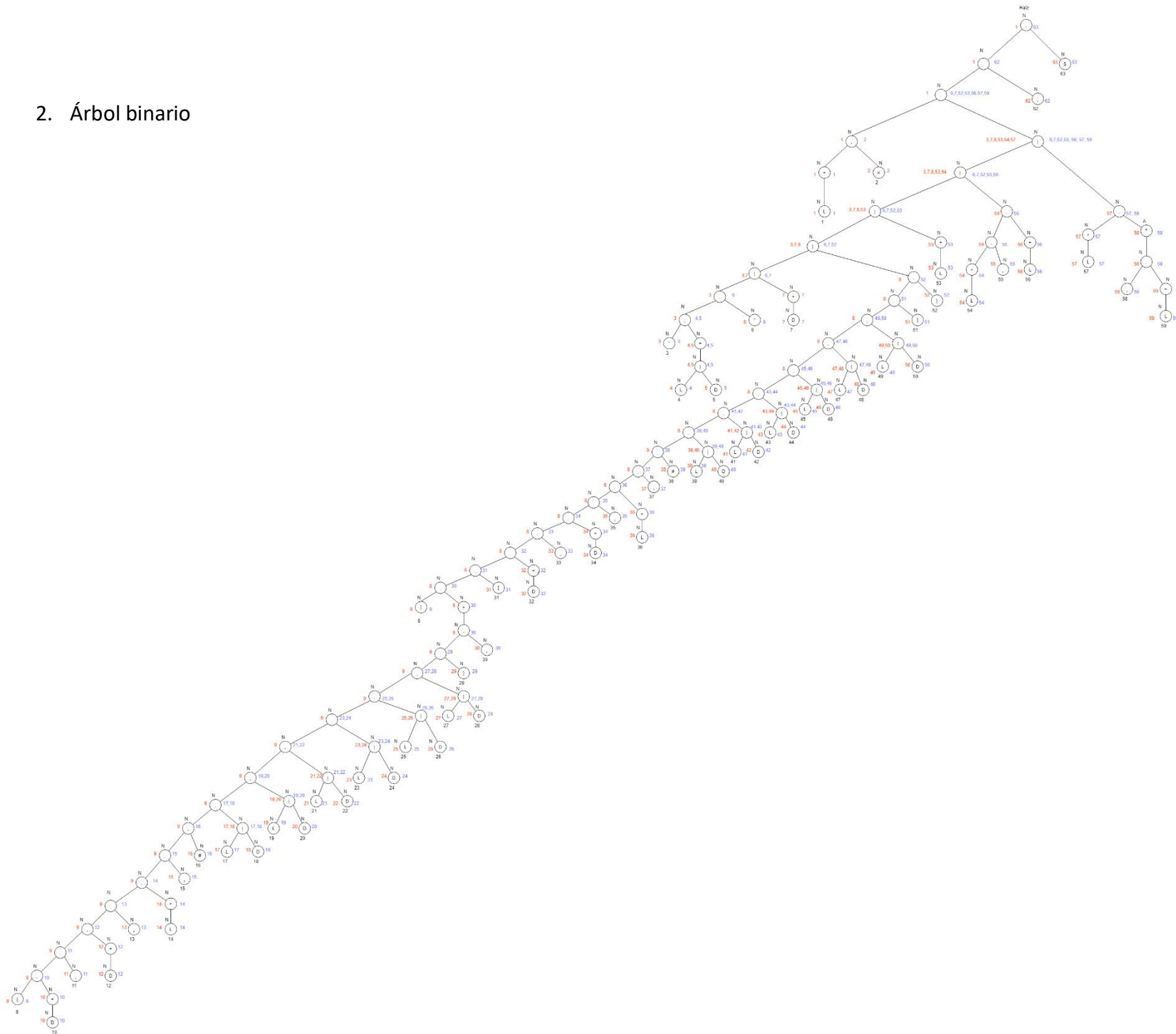
Token	Expresión regular
Título	L+=“(L D I)+”;
Ancho	L+=D+;
Alto	L+=D+;
Filas	L+=D+;
Columnas	L+=D+;
Celdas	L+=“{([D+,D+,L+,#(L D)(L D)(L D)(L D)(L D)(L D)],)+[D+,D+,L+,#(L D)(L D)(L D)(L D)(L D)(L D)]}”;
Filtros	L+=L+ (, L+)*

Las expresiones regulares comparten el mismo inicio y fin, es por ello que se unieron en una sola expresión regular para crear el árbol binario.

1. ER:

L+=“(“L|D|I)+”|D+|“{([D+,D+,L+,#(L|D)(L|D)(L|D)(L|D)(L|D)(L|D)],)+[D+,D+,L+,#(L|D)(L|D)(L|D)(L|D)(L|D)(L|D)]}”|L+(,L+)*”;\$

2. Árbol binario

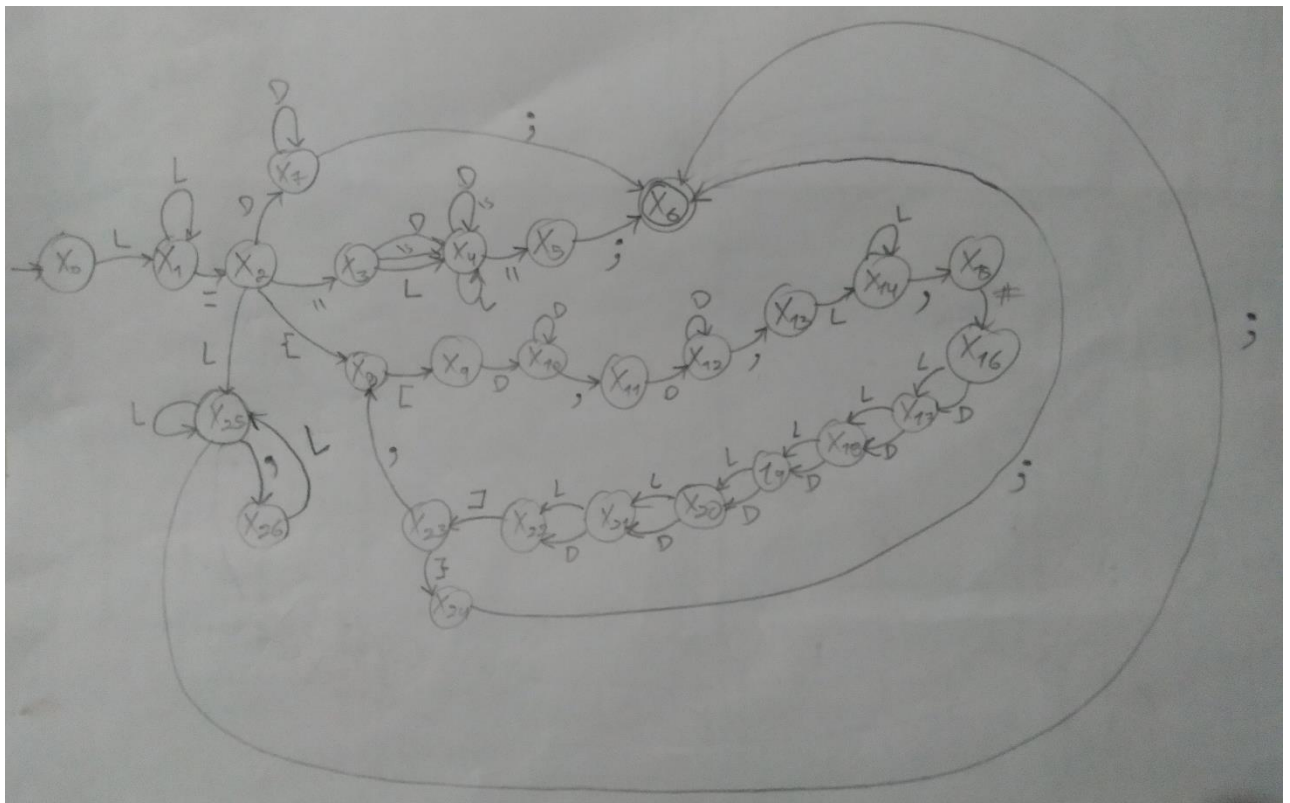


3. Tabla de Follow.

Hoja	Valor	Follow

4. Tabla de transiciones.

5. Autómata



Luego de tener claro el autómata, se utilizó tres archivos dentro del proyecto para poder manipular los datos se describen a continuación:

Tabla II

Componentes del archivo lexemas.

Lexemass.py	
Clase	lexema: se utiliza para crear los objetos de tipo lexema con los atributos fila y columna. Los métodos que posee son verLexe, verFila y verColumna los cuales son utilizados para obtener la información de cada alumno.

Fuente: elaboración propia.

Tabla III

Componentes del archivo ImagenObjs.

ImagenObj.py	
Librerías utilizadas	Graphviz: esta librería permite crear archivos de texto plano en el lenguaje DOT, el cual luego es renderizado para generar imágenes en formato PNG.
Clase	imagenObjeto: se utiliza para crear los objetos de tipo imagen con los atributos titulo, ancho, alto, filas, columnas, celdas, filtros, celdasSep, imagen.
Métodos	<p>Los métodos que posee son:</p> <p>verTitulo: retornar el nombre de la imagen.</p> <p>verAncho: retornar el ancho de la imagen.</p> <p>verAlto: retornar el alto de la imagen.</p> <p>verFilas: retornar el número de filas de la imagen.</p> <p>verColumnas: retornar el número de columnas de la imagen.</p> <p>verCeldas: retornar en una sola cadena de texto todas las celdas que componen la imagen.</p> <p>verFiltros: retornar los filtros que se aplicarán a la imagen.</p> <p>separarCeldas: esta consta de tres fases, en la cual se toma como base la información en el atributo celdas, la fase 1 separa las celdas que se encuentran en una sola cadena de texto y las agrega a una lista con el nombre celdasSep, la separación la hace cada cuatro comas. La fase 2 le da dimensione de matriz al atributo imagen, en la cual se usa los datos de número de filas y columnas. La fase 3 recorre la lista de celdasSep y extrae los datos de cada celda por separado donde se verifica si el estado es TRUE, luego se analiza el</p>

	<p>tipo de filtro y según lo solicitado se hace llamado a los métodos de crear grafos y archivos HTML.</p> <p>crearGrafo: escribe un archivo de texto plano con extensión DOT haciendo uso de la librería Graphviz, dentro del archivo se crea una tabla según el tipo de filtro que se debe aplicar a la imagen, esto se toma de los parámetros del método.</p> <p>crearHTML: escribe dos archivos de texto plano, el primero con extensión CSS y el segundo con extensión HTML, donde se crea el estilo del sitio web en el CSS y en el HTML se inserta la imagen según nombre del filtro el cual es tomado de los parámetros del método.</p> <p>verCeldasSeparadas: retorna la lista de los elementos almacenados dentro del atributo celdasSep.</p>
--	---

Fuente: elaboración propia.

En el archivo principal se crea la interfaz gráfica, esta inicialmente solo tiene habilitado un botón, el resto se activarán según se analicen los datos.

Tabla IV

Componentes del archivo main.

main.py	
Librerías utilizadas	<p>tkinter: esta librería permite crear interfaz gráfica, con la cual el usuario puede interactuar a través de marcos, paneles, botones, etiquetas, listas desplegables y cuadros de mensaje.</p> <p>PIL: la librería permite renderizar imágenes dentro de la interfaz gráfica, las cuales luego son mostradas dentro de una etiqueta.</p> <p>re: el uso de esta librería es exclusivamente para verificar la extensión del archivo de texto plano, la cual debe ser únicamente “.pxla”.</p> <p>webbrowser: con la librería se puede manipular el navegador web que se encuentre por defecto, para que se abra automáticamente con el archivo HTML que se le indique.</p> <p>imagenObj: se utiliza para crear objetos de tipo imagenObjeto.</p> <p>lexemas: se utiliza para crear objetos de tipo lexema.</p>
Variables globales	<p>rutaArchivo: almacena la ruta del archivo cargado.</p> <p>archivoPXLA = almacena toda a información del archivo cargado.</p> <p>imagenes: es una lista en la cual se almacenan las imágenes que</p>

	<p>fueron separadas por las cuatro @.</p> <p>imgObjetos: es una lista que almacena las imágenes convertidas en objetos de tipo imagenObjeto.</p> <p>reservadas: es una lista que posee las palabras válidas dentro de los atributos de las imágenes, la cual es utilizada para comparar los lexemas obtenidos del autómata.</p> <p>lexemaValidos: es una lista que almacena los lexemas aceptados, luego de que fueron comparados en la lista de reservadas.</p> <p>lexemasError: es una lista que almacena los lexemas no aceptados, luego de que fueron comparados en la lista de reservadas.</p>
Funciones	<p>Los métodos que posee son:</p> <p>esLetra: convierten un carácter a su equivalente entero ASCII y verifica si se encuentra entre el rango de 65 y 90, o 97 y 122, o es igual a 165 o 164, retorna TRUE o FALSE según el caso.</p> <p>esNumero: convierten un carácter a su equivalente entero ASCII y verifica si se encuentra entre el rango de 48 y 57, retorna TRUE o FALSE según el caso.</p> <p>imprimible: convierten un carácter a su equivalente entero ASCII y verifica si se encuentra entre el rango de 128 y 239, retorna TRUE o FALSE según el caso.</p> <p>analizar: recorre carácter a carácter toda la información cargada del archivo de texto plano, donde se hace uso del autómata con sus 26 estados, donde únicamente el estado de aceptación es el estado 6, en este estado se valida que el lexema sea valido y se eliminan los títulos de las cadenas para solo dejar los datos de la imagen y agregarlos a la lista de objetos lexemasValidos y la lista de objetos imagenObjeto, en caso contrario los datos son enviados a la lista de objetos lexemasError.</p> <p>analizarImagenes: recorre la lista llamada imagenes para poder aplicar a cada imagen la función analizar, luego habilita el resto de botones que hacen falta en el entorno gráfico. Por último, se recorre la lista de imgObjetos para poder crear a cada uno su archivo DOT e imagen, finalizado el proceso se notifica al usuario a través de un cuadro de texto.</p> <p>habilitarBotones1: habilita en el entorno gráfico los botones cargar, analizar archivo y ver reportes.</p> <p>verReporte: crea un archivo HTML en el cual se describe la información de la lista lexemasValidos, donde se especifica la fila y columna donde se localizó el lexema dentro de archivo de texto</p>

	<p>plano, como también muestra los errores léxicos.</p> <p>verImagen: habilita los botones de ORIGINAL, MIRRORX, MIRROR, DOUBLEMIRROR y llama la función verOriginal si se seleccionó un elemento de la lista desplegable, en caso contrario le notifica al usuario que debe elegir una imagen.</p> <p>verOriginal: muestra en la etiqueta utilizada como visor la imagen original de las celdas analizadas anteriormente.</p> <p>verMirrorX: muestra la imagen en el visor correspondiente al filtro, si este se encuentra disponible según la información cargada en el archivo de texto plano, en caso contrario lo notifica al usuario.</p> <p>verMirrorY: muestra la imagen en el visor correspondiente al filtro, si este se encuentra disponible según la información cargada en el archivo de texto plano, en caso contrario lo notifica al usuario.</p> <p>verMirrorX: muestra la imagen en el visor correspondiente al filtro, si este se encuentra disponible según la información cargada en el archivo de texto plano, en caso contrario lo notifica al usuario.</p> <p>habilitarBotones2: habilita los botones cargar y analizar archivo, como también la lista desplegable donde el usuario seleccionar el título de la imagen que desea ver.</p> <p>separarImagenes: toma toda la información del archivo de texto y separa la cadena cada vez que encuentra cuatro arrobas, cuando se localiza las arrobas la cadena de texto es enviada a la lista llamada imagenes.</p> <p>abrirArchivo: verifica que el archivo que selecciono el usuario sea con extensión “.pxla”, si esto es correcto entonces extrae la información y al final de la cadena le agrega cuatro arrobas para después ser almacenado en la variable archivoPXLA y llamar a la función habilitarBotones1, en caso contrario notifica al usuario que el archivo no es válido.</p>
--	---

Fuente: elaboración propia.

Observaciones

- Las carpetas docu, html e imágenes no deben ser eliminadas, ya que son utilizadas por el software, de igual forma no se debe borrar la imagen llamada no-imagen que se encuentra dentro de la carpeta docu.
- Los archivos DOT e imágenes con sus respectivos filtros se generan dentro de la carpeta imágenes.
- Los reportes HTML se generan dentro de la carpeta con el nombre html.